

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Takeyuki UEDA

Application No.:

Group Art Unit:

Filed: August 26, 2003

Examiner:

For: MOBILE TERMINAL

**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN
APPLICATION IN ACCORDANCE
WITH THE REQUIREMENTS OF 37 C.F.R. § 1.55**

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 37 C.F.R. § 1.55, the applicant(s) submit(s)
herewith a certified copy of the following foreign application:

Japanese Patent Application No(s). 2002-255729

Filed: August 30, 2002

It is respectfully requested that the applicant(s) be given the benefit of the foreign filing
date(s) as evidenced by the certified papers attached hereto, in accordance with the
requirements of 35 U.S.C. § 119.

Respectfully submitted,

STAAS & HALSEY LLP

Date: August 26, 2003

By: 

Paul I. Kravetz
Registration No. 35,230

1201 New York Ave, N.W., Suite 700
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日
Date of Application:

2002年 8月30日

出 願 番 号
Application Number:

特願2002-255729

[ST.10/C]:

[JP2002-255729]

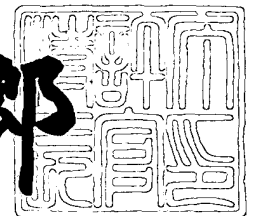
出 願 人
Applicant(s):

富士通株式会社

2002年12月27日

特 許 庁 長 官
Commissioner,
Japan Patent Office

太田信一郎



出証番号 出証特2002-3103092

【書類名】 特許願

【整理番号】 0252046

【提出日】 平成14年 8月30日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 3/00 654

【発明の名称】 携帯端末及びアプリケーションプログラム

【請求項の数】 10

【発明者】

 【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

 【氏名】 上田 健之

【特許出願人】

 【識別番号】 000005223

 【氏名又は名称】 富士通株式会社

【代理人】

 【識別番号】 100092978

 【弁理士】

 【氏名又は名称】 真田 有

 【電話番号】 0422-21-4222

【手数料の表示】

 【予納台帳番号】 007696

 【納付金額】 21,000円

【提出物件の目録】

 【物件名】 明細書 1

 【物件名】 図面 1

 【物件名】 要約書 1

 【包括委任状番号】 9704824

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 携帯端末及びアプリケーションプログラム

【特許請求の範囲】

【請求項 1】 同種のグラフィカル・ユーザ・インターフェース（以下、G U I という）部品を複数個含む G U I 画面を表示させるアプリケーションプログラムの G U I 画面の画面テーマを切り替える処理を実行する制御部と、

画面テーマを定義する画面テーマ定義データを格納する記憶部とを備え、

前記制御部が、

切替後の画面テーマの画面テーマ定義データを前記記憶部から読み出し、前記アプリケーションプログラムのために確保されたメモリ領域に記憶させる切替時画面テーマ読出処理ステップと、

前記アプリケーションプログラムのメモリ領域に記憶されている画面テーマ定義データを用いて前記 G U I 画面に含まれる同種の複数個の G U I 部品を描画する G U I 画面描画処理ステップとを含む処理を実行するように構成されることを特徴とする、携帯端末。

【請求項 2】 前記制御部が、前記切替時画面テーマ読出処理ステップにおいて、前記記憶部から前記 G U I 画面に含まれる G U I 部品の画面テーマ定義データのみを読み出し、前記アプリケーションプログラムのメモリ領域に記憶させる処理を実行するように構成されることを特徴とする、請求項 1 記載の携帯端末。

【請求項 3】 前記制御部が、

前記アプリケーションプログラムの起動時に、デフォルトとして設定された画面テーマの画面テーマ定義データを前記記憶部から読み出し、前記アプリケーションプログラムのメモリ領域に記憶させるアプリ起動時画面テーマ読出処理ステップを実行するように構成され、

前記切替時画面テーマ読出処理ステップにおいて、切替後の画面テーマの画面テーマ定義データを前記記憶部から読み出し、前記アプリケーションプログラムのメモリ領域に記憶されている画面テーマ定義データを更新する処理を実行するように構成されることを特徴とする、請求項 1 又は 2 記載の携帯端末。

【請求項 4】 前記制御部が、前記切替時画面テーマ読出処理ステップにおいて、オペレーティングシステムによって管理される画面テーマに関する情報を用いて前記記憶部から画面テーマ定義データを読み出す処理を実行するように構成されることを特徴とする、請求項 1～3 のいずれか 1 項に記載の携帯端末。

【請求項 5】 前記制御部が、
前記記憶部に格納されている画面テーマ定義データを読み出し、オペレーティングシステムのために確保されたメモリ領域に展開する処理を実行するように構成され、

前記記憶部から前記画面テーマ定義データを読み出すのに代えて、前記記憶部から読み出され、前記オペレーティングシステムのメモリ領域に展開されている前記画面テーマ定義データを読み出す処理を実行するように構成されることを特徴とする、請求項 1～4 のいずれか 1 項に記載の携帯端末。

【請求項 6】 前記制御部が、オペレーティングシステムによって管理される画面テーマに関する情報が書き換えられた場合に、前記切替時画面テーマ読出処理ステップ及び前記 G U I 画面描画処理ステップを実行するように構成されることを特徴とする、請求項 1～5 のいずれか 1 項に記載の携帯端末。

【請求項 7】 前記制御部が、
画面テーマの設定に用いる画面テーマ設定アプリケーションプログラムにしたがって、画面テーマ決定指令に応じて、オペレーティングシステムによって管理される画面テーマに関する情報を切替後の画面テーマに関する情報に書き換える処理を実行するように構成され、

前記オペレーティングシステムにしたがって、前記画面テーマに関する情報が書き換えられた場合に、前記アプリケーションプログラムに対して画面テーマ変更通知を行なう処理を実行するように構成され、

前記アプリケーションプログラムにしたがって、前記オペレーティングシステムからの画面テーマ変更通知を受けた場合に、前記切替時画面テーマ読出処理ステップ及び前記 G U I 画面描画処理ステップを実行するように構成されることを特徴とする、請求項 1～5 のいずれか 1 項に記載の携帯端末。

【請求項 8】 前記制御部が、画面テーマの設定に用いる画面テーマ設定ア

アプリケーションプログラムのGUI画面に含まれる画面テーマリストの選択位置に応じて、前記画面テーマ設定アプリケーションプログラムのGUI画面の画面テーマを切り替える処理を実行するように構成されることを特徴とする、請求項1～7のいずれか1項に記載の携帯端末。

【請求項9】 前記制御部が、画面テーマの設定に用いる画面テーマ設定アプリケーションプログラムのGUI画面に含まれる画面テーマリストの選択位置に応じて、前記画面テーマ設定アプリケーションプログラムのGUI画面上にビットマップイメージを表示させる処理を実行するように構成されることを特徴とする、請求項1～7のいずれか1項に記載の携帯端末。

【請求項10】 同種のグラフィカル・ユーザ・インターフェース（以下、GUIという）部品を複数個含むGUI画面の画面テーマを切り替える画面テーマ切替処理をコンピュータに実行させるためのアプリケーションプログラムであって、

前記画面テーマ切替処理が、

切替後の画面テーマの画面テーマ定義データを画面テーマリソースファイルから読み出し、前記アプリケーションプログラムのために確保されたメモリ領域に記憶させる切替時画面テーマ読出処理ステップと、

前記アプリケーションプログラムのメモリ領域に記憶されている画面テーマ定義データを用いて前記GUI画面に含まれる同種の複数個のGUI部品を描画するGUI画面描画処理ステップとを含むことを特徴とする、アプリケーションプログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、アプリケーションプログラム（以下、アプリ、アプリケーションともいう）によって表示されるGUI（グラフィカル・ユーザ・インターフェース）部品からなるGUI画面を介してユーザが種々の操作を行なうことができる、携帯端末及びアプリケーションプログラムに関する。

【0002】

【従来の技術】

近年、パーソナルコンピュータにおいて、G U I 画面を全体的に統一されたテーマ（画面テーマ）で表現し、その画面テーマを切り替える機能（テーマ機能）を備えるものが提供されている。

このような機能を持つものとしては、例えば、マイクロソフト社のウィンドウズ（登録商標）のデスクトップ・テーマやリアルオーディオのスキンなどを挙げることができる。

【0 0 0 3】

【発明が解決しようとする課題】

ところで、近年、例えば携帯電話や携帯情報端末（P D A ; Personal Digital Assistants）などの携帯端末の高機能化が進んでおり、パーソナルコンピュータに実装されている機能と同様の機能（例えば、上述のテーマ機能）を実装することが望まれている。

【0 0 0 4】

しかしながら、携帯端末では、携帯性を考慮しなければならず、サイズや重さなどの制約を受けるため、パーソナルコンピュータのように、処理速度の速いC P Uを用いたり、メモリの容量を多くしたりすることができない。

このため、このような制約の下、パーソナルコンピュータに実装されている機能と同様の機能を携帯端末に実装するには、何らかの工夫が必要になる。

【0 0 0 5】

特に、携帯端末において、上述のテーマ機能を実装するためには、画面テーマが高速で切り替わるようにすること、画面テーマを切り替えるために使用するメモリ領域をできるだけ少なくすることが重要になる。なお、これらの点は、上述の機能を備えるパーソナルコンピュータなどにおいても必要なことである。

本発明は、このような課題に鑑み創案されたもので、アプリケーションの画面が全体的に統一されたテーマ（画面テーマ）で表現されるようにし、このような画面テーマが高速で切り替わるようにした携帯端末及び画面テーマが高速で切り替わるようにするためのアプリケーションプログラムを提供することを目的とする。

【0006】

また、本発明は、できるだけ少ないメモリ領域を使用して画面テーマの切り替え（特に、高速切替）を行なえるようにすることも目的とする。

【0007】

【課題を解決するための手段】

このため、本発明の携帯端末は、同種のグラフィカル・ユーザ・インターフェース（以下、G U I という）部品を複数個含むG U I 画面を表示させるアプリケーションプログラムの画面テーマを切り替える処理を実行する制御部と、画面テーマを定義する画面テーマ定義データを格納する記憶部とを備え、制御部が、切替後の画面テーマの画面テーマ定義データを記憶部から読み出し、アプリケーションプログラムのために確保されたメモリ領域に記憶させる切替時画面テーマ読出処理ステップと、アプリケーションプログラムのメモリ領域に記憶されている画面テーマ定義データを用いてG U I 画面に含まれる同種の複数個のG U I 部品を描画するG U I 画面描画処理ステップとを含む処理を実行するように構成されることを特徴としている（請求項1）。

【0008】

好ましくは、制御部を、切替時画面テーマ読出処理ステップにおいて、記憶部からG U I 画面に含まれるG U I 部品の画面テーマ定義データのみを読み出し、アプリケーションプログラムのメモリ領域に記憶させる処理を実行するように構成する（請求項2）。

また、制御部を、アプリケーションプログラムの起動時に、デフォルトとして設定された画面テーマの画面テーマ定義データを記憶部から読み出し、アプリケーションプログラムのメモリ領域に記憶させるアプリ起動時画面テーマ読出処理ステップを実行するように構成し、切替時画面テーマ読出処理ステップにおいて、切替後の画面テーマの画面テーマ定義データを記憶部から読み出し、アプリケーションプログラムのメモリ領域に記憶されている画面テーマ定義データを更新する処理を実行するように構成するのも好ましい（請求項3）。

【0009】

また、制御部を、切替時画面テーマ読出処理ステップにおいて、オペレーティ

ングシステムによって管理される画面テーマに関する情報を用いて記憶部から画面テーマ定義データを読み出す処理を実行するように構成するのも好ましい（請求項4）。

また、制御部を、記憶部に格納されている画面テーマ定義データを読み出し、オペレーティングシステムのために確保されたメモリ領域に展開する処理を実行するように構成し、記憶部から画面テーマ定義データを読み出すのに代えて、記憶部から読み出され、オペレーティングシステムのメモリ領域に展開されている画面テーマ定義データを読み出す処理（切替時画面テーマ読出処理ステップ、アプリ起動時画面テーマ読出処理ステップ）を実行するように構成するのも好ましい（請求項5）。

【0010】

また、制御部を、オペレーティングシステムによって管理される画面テーマに関する情報が書き換えられた場合に、切替時画面テーマ読出処理ステップ及びGUI画面描画処理ステップを実行するように構成するのも好ましい（請求項6）。

また、制御部を、画面テーマの設定に用いる画面テーマ設定アプリケーションプログラムにしたがって、画面テーマ決定指令に応じて、オペレーティングシステムによって管理される画面テーマに関する情報を切替後の画面テーマに関する情報に書き換える処理を実行するように構成し、オペレーティングシステムにしたがって、画面テーマに関する情報が書き換えられた場合に、アプリケーションプログラムに対して画面テーマ変更通知を行なう処理を実行するように構成し、アプリケーションプログラムにしたがって、オペレーティングシステムからの画面テーマ変更通知を受けた場合に、切替時画面テーマ読出処理ステップ及びGUI画面描画処理ステップを実行するように構成するのも好ましい（請求項7）。

【0011】

また、制御部を、画面テーマの設定に用いる画面テーマ設定アプリケーションプログラムのGUI画面に含まれる画面テーマリストの選択位置に応じて、画面テーマ設定アプリケーションプログラムのGUI画面の画面テーマを切り替える処理を実行するように構成するもの好ましい（請求項8）。

また、制御部を、画面テーマの設定に用いる画面テーマ設定アプリケーションプログラムのG U I画面に含まれる画面テーマリストの選択位置に応じて、画面テーマ設定アプリケーションプログラムのG U I画面上にビットマップイメージを表示させる処理を実行するように構成するのも好ましい（請求項9）。

【0012】

本発明のアプリケーションプログラムは、同種のグラフィカル・ユーザ・インターフェース（以下、G U Iという）部品を複数個含むG U I画面の画面テーマを切り替える画面テーマ切替処理をコンピュータに実行させるためのアプリケーションプログラムであって、画面テーマ切替処理が、切替後の画面テーマの画面テーマ定義データを画面テーマリソースファイルから読み出し、アプリケーションプログラムのために確保されたメモリ領域に記憶させる切替時画面テーマ読出処理ステップと、アプリケーションプログラムのメモリ領域に記憶されている画面テーマ定義データを用いてG U I画面に含まれる同種の複数個のG U I部品を描画するG U I画面描画処理ステップとを含むことを特徴としている。

【0013】

【発明の実施の形態】

以下、図面により、本発明の一実施形態にかかる携帯端末、アプリケーションプログラム及び同プログラムを格納したコンピュータ読取可能な記憶媒体について説明する。

本実施形態では、携帯端末としての携帯電話を例に、図1～図10を参照しながら説明する。

【0014】

まず、携帯電話の構成について、図1～図10を参照しながら説明する。

図1に示すように、携帯電話1は、出力部（表示部）としてのディスプレイ2と、入力部としての操作部（例えば操作ボタンなど）3と、例えばC P Uやメモリ（揮発性メモリ；R A M）などからなる制御部（コンピュータ）4と、例えばフラッシュメモリ（不揮発性メモリ；R O M）などの記憶部5とを備えて構成される。なお、当然のことながら、無線通信に必要なもの（例えばアンテナや信号処理部等）も備えられている。

【 0 0 1 5 】

そして、記憶部 5 には、オペレーティングシステム（以下、OS ともいう）、種々のアプリケーションプログラム（以下、アプリ、アプリケーションともいう）が格納されており、これらの OS やアプリが、制御部 4 のメモリ上にロードされ、CPU によって実行されることで、携帯電話 1 の所望の機能が実現されるようになっている。

【 0 0 1 6 】

特に、本実施形態では、OS は、システムの信頼性、安全性を確保すべく、あるプロセスから他のプロセスへ高速で切り替えることで、複数のプロセスを見かけ上同時に実行できるマルチプロセス機能を有するマルチプロセス OS としている。

これは、単一のメモリ領域上で全てのアプリケーションやカーネル（OS の基本機能）が動作するモノリシック OS では、一つのプログラムでトラブル（例えばメモリアクセスの誤りなど）が生じた場合にはシステム全体が停止してしまうことになるからである。

【 0 0 1 7 】

これに対し、本実施形態のマルチプロセス OS は、プロセス毎に使用できるメモリ領域を制限するメモリ保護機能を備えており、OS やアプリなどのプログラム（プロセス）がそれぞれ異なるメモリ領域を使用するようになっており、一つのプログラム（プロセス）でトラブル（例えばメモリアクセスの誤りなど）が生じた場合であっても、OS や他のアプリケーションに影響を与えないようになっている。

【 0 0 1 8 】

このように、携帯電話 1 においてマルチプロセス OS を用いる場合、オーバーヘッドを抑えることが重要になる。つまり、例えばパーソナルコンピュータのように処理速度が速い CPU を用いることができれば、オーバーヘッドはそれほど問題とならないが、携帯電話では、携帯性を考慮しなければならず、例えばパーソナルコンピュータのように処理速度の速い CPU を用いることはできず、処理速度の遅い CPU を用いることになるため、オーバーヘッドを抑えることが重要

になる。

【 0 0 1 9 】

また、各アプリは、ディスプレイ 2 上にグラフィカル・ユーザ・インターフェース（以下、G U I という）部品からなる G U I 画面を表示させるものとして構成される。そして、ユーザがディスプレイ 2 上に表示された G U I 画面を見ながら操作部 3 を操作することで、制御部 4 に対して各種の指令を入力することができるようになっている。

【 0 0 2 0 】

ところで、このような G U I 画面を表示させるアプリを使用可能な携帯電話 1 において、ユーザの好みや気分によって G U I 画面の雰囲気を変えることができるように、各アプリの G U I 画面が統一された所望のテーマ（画面テーマ）で表現されるようにし、ユーザが任意に G U I 画面の画面テーマを切り替えることができるようにしたい。

【 0 0 2 1 】

つまり、G U I 画面を表示させるアプリを使用可能な携帯電話において、G U I 画面に例えば装飾的な高級感，機械的な機能感，文房具的な親近感などの雰囲気をもたせるべく、各アプリの G U I 画面を構成する複数の G U I 部品の例えば色彩，立体感の有無，ボーダーの有無などを統一することで、G U I 画面が所望のテーマ（画面テーマ）で表現されるようにし、ユーザが任意に G U I 画面の画面テーマを切り替えることができるようにしたい。

【 0 0 2 2 】

このような G U I 画面の画面テーマの切替機能を携帯電話 1 に実装するには、G U I 画面の画面テーマが切り替えられた場合に、各アプリの G U I 画面の画面テーマが高速で切り替わるようにすることが重要である。

特に、上述のように、メモリ保護機能を有するマルチプロセス O S を備える携帯電話 1 において、各アプリの G U I 画面の画面テーマが高速で切り替わるようにするためには、プロセスをまたいでメモリアクセスやコンテキストスイッチングをできるだけ減らして、オーバーヘッドを抑えることが重要になる。

【 0 0 2 3 】

このため、本実施形態の携帯電話 1 では、以下の (1) ～ (3) に示すようなシステム構成としている。

(1) G U I 部品 (コントロール) の画面テーマにかかわる属性 (G U I 画面を統一されたテーマで表現するために必要な属性) を G U I 部品の種類毎に定義した画面テーマ定義データを格納して画面テーマリソースファイルを作成する。ここでは、画面テーマ毎に複数の画面テーマリソースファイルを作成する。

【 0 0 2 4 】

そして、これらの画面テーマ毎の画面テーマリソースファイルを記憶部 (例えばフラッシュメモリ) 5 に格納し、O S のファイルシステムによって管理する。

具体的には、画面テーマ毎に作成される複数の画面テーマリソースファイルには、携帯電話 1 で用いることができる全種類の G U I 部品の属性データ、例えば背景色、背景ビットマップ、ボーダーの有無などのデータを圧縮して格納しておく。

(2) 一方、上述の (1) のように、画面テーマを定義した画面テーマ定義データを画面テーマリソースファイルに格納し、O S のファイルシステムによって管理されるようにすると、各アプリの G U I 画面の画面テーマを切り替える際に、G U I 画面を再描画するのに、G U I 画面を構成する G U I 部品の個数分だけ、画面テーマリソースファイルを参照しなければならない。

【 0 0 2 5 】

この場合、アプリと O S のカーネルとの間で G U I 部品の個数分だけコンテキストスイッチングが生じるため、オーバーヘッドとなり、再描画が大変遅いものになってしまう (例えばスリープが生じてしまう) ため、現実的ではない。なお、画面テーマ定義データを O S のために確保されているメモリ領域に展開しておく場合も、コンテキストスイッチングは生じることに変わりはないため、同様である。

【 0 0 2 6 】

このため、本実施形態では、同種の G U I 部品を複数個含む G U I 画面を表示させるアプリの画面テーマの高速切替を実現すべく、以下のように構成している。

つまり、アプリのGUI画面の画面テーマを切り替える処理を実行する制御部4が、GUI画面の画面テーマの切替時に、切替後の画面テーマの画面テーマ定義データを画面テーマリソースファイルから読み出して、アプリのために確保されたメモリ領域に記憶させ、アプリのメモリ領域に記憶されている画面テーマ定義データを用いてGUI画面に含まれる複数個のGUI部品を描画するようにして、アプリとOSのカーネルとの間で生じるコンテキストスイッチングの回数をできるだけ減らし、オーバーヘッドを抑えることで、GUI画面の画面テーマの高速切替を実現している。

【0027】

特に、本実施形態では、制御部4が、アプリのGUI画面に含まれるGUI部品の画面テーマ定義データのみを読み出してアプリのメモリ領域に記憶させるようにして、GUI部品の画面テーマ定義データを記憶させるのに必要なメモリ領域を少なくし、メモリ容量の少ない携帯電話において画面テーマの高速切替を実現できるようにしている。

【0028】

また、本実施形態では、制御部4が、アプリの起動時に、デフォルトとして設定された画面テーマの画面テーマ定義データを画面テーマリソースファイルから読み出して、アプリのメモリ領域に記憶させるようにし、画面テーマの切替時には、アプリのメモリ領域に記憶されている画面テーマ定義データを更新する（書き換える）ようにして、さらなる画面テーマの切り替えの高速化を図るとともに、画面テーマを切り替えるのに必要なメモリ領域ができるだけ少なくて済むようにしている。

【0029】

なお、本実施形態では、画面テーマリソースファイルから画面テーマ定義データを読み出しているが、これに限られるものではなく、制御部4が、予め画面テーマリソースファイルに格納されている画面テーマ定義データを、OSのために確保されたメモリ領域に展開しておき、このOSのメモリ領域に展開されている画面テーマ定義データを読み出すようにしても良い。この場合、例えば画面テーマ定義データをエクセルシートによって作成しておき、OS起動時にOSのメモ

り領域上に展開されるようにすれば良い。

【 0 0 3 0 】

以下、このような画面テーマ切替処理を制御部（コンピュータ）4 に実行させるためのアプリケーションプログラムの構成について、図 1 を参照しながら説明する。なお、ここではアプリケーションプログラムをオブジェクト指向プログラミングによって構成する場合を例に説明する。

ここでは、アプリ毎に画面テーマクラスを設け、この画面テーマクラスが画面テーマ定義データを取り込んで画面テーマインスタンスを生成するようにし、アプリが G U I 画面を描画する際に、同種の複数個の G U I 部品を描画するのに、同一の画面テーマインスタンスをキャッシュメモリのように利用するようにしている。

【 0 0 3 1 】

具体的には、本アプリケーションプログラムは、図 1 に示すように、G U I 画面を描画するためのクラスとして G U I 画面クラス（アプリ画面クラス）を備える。

ここで、G U I 画面クラスは、G U I 画面を構成する各 G U I 部品を描画するための G U I 部品クラス（例えば、ボタンクラス、ウィンドウクラス、アンテナ・電池残量等クラスなど）を集約したものとして構成される。なお、G U I 画面クラスは、各アプリ毎に、各アプリの G U I 画面を構成する G U I 部品を描画するのに必要な G U I 部品クラスのみを集約したものとして構成している。

【 0 0 3 2 】

特に、本実施形態では、同種の G U I 部品の構成要素を、G U I 部品毎に固有の構成要素（固有構成要素）と、G U I 部品に共通の構成要素（共通構成要素）とに分け、このうち、G U I 部品の固有構成要素を、G U I 部品クラスの属性として持たせている。例えば、ボタンクラスは、固有構成要素として、例えばボタンの描画位置とサイズ、文字列などの属性を持つものとして構成している。

【 0 0 3 3 】

一方、本実施形態では、G U I 画面クラスのほかに、G U I 画面の画面テーマを形成するためのクラスとして G U I 画面テーマクラスを設けている。

ここで、G U I 画面テーマクラスは、各G U I 部品の画面テーマを形成するためのG U I 部品画面テーマクラス（例えば、ボタン画面テーマクラス、ウィンドウ画面テーマクラス、アンテナ・電池残量等画面テーマクラスなど）を集約したものとして構成される。本実施形態では、G U I 画面テーマクラスは、アプリのG U I 画面を構成するために用いることができる全てのG U I 部品のG U I 部品画面テーマクラスを集約したものとして構成している。

【 0 0 3 4 】

特に、本実施形態では、G U I 部品の共通構成要素が画面テーマを形成しうる構成要素となりうるため、この共通構成要素（テーマ形成構成要素）を、G U I 部品画面テーマクラスの属性として持たせている。例えば、ボタン画面テーマクラスは、テーマ形成構成要素として、例えばボタンの背景色、フォントフェイス、枠線の有無などの属性を持つものとして構成している。

【 0 0 3 5 】

そして、本実施形態では、アプリケーションの起動時に、G U I 画面クラス、各G U I 部品クラス、G U I 画面テーマクラス、各G U I 部品画面テーマクラスのメソッドが制御部4によって実行されて、図1に示すように、各G U I 部品クラス（例えばボタンクラス）からそれぞれG U I 部品インスタンス（例えばボタン1インスタンス、ボタン2インスタンス）が生成されるとともに、各G U I 部品画面テーマクラス（例えばボタン画面テーマクラス）からそれぞれG U I 部品画面テーマインスタンス（例えばボタン画面テーマインスタンス）が生成されるようになっている。

【 0 0 3 6 】

具体的には、G U I 部品画面テーマクラスのメソッドが制御部4によって実行され、O Sによって管理されている現在の画面テーマのリソースファイル名（現在の画面テーマに関する情報）を用いて画面テーマリソースファイルから画面テーマ定義データが読み出され、この画面テーマ定義データを用いてG U I 部品画面テーマインスタンスが生成されるようになっている。そして、生成されたG U I 部品画面テーマインスタンスを、アプリケーションプログラムのために確保されたメモリ領域に記憶し、G U I 部品画面テーマインスタンスを記憶したメモリ

領域のアドレス（ポインタ）をGUI部品クラスに通知するようになっている。

【0037】

例えば、ボタン画面テーマクラスのメソッドが制御部4によって実行されると、図1に示すように、ボタン画面テーマインスタンスが生成されることになる。

特に、本実施形態では、GUI部品画面テーマクラスのメソッドは、GUI部品クラスからの画面テーマ要求メッセージに応じて、制御部4によって実行されるようになっているため、各アプリのGUI画面を構成するのに必要なGUI部品のGUI部品画面テーマインスタンスのみが生成されることになる。

【0038】

なお、アプリケーションの起動時には、デフォルトとして設定されている画面テーマリソースファイル名を現在の画面テーマリソースファイル名として用いる。

一方、GUI部品クラスのメソッドが制御部4によって実行され、GUI部品インスタンスが生成されるようになっている。そして、生成されたGUI部品インスタンスを、アプリケーションプログラムのために確保されたメモリ領域に記憶し、そのメモリ領域のアドレス（ポインタ）をGUI部品クラスに通知する。

【0039】

例えば、ボタンクラスのメソッドが制御部4によって実行されると、図1に示すように、ボタン1インスタンスと、ボタン2インスタンスとが生成されることになる。そして、これらのボタンインスタンスは、それらが記憶されたメモリ領域のアドレス（ポインタ）をボタンクラスに通知する。

そして、GUI画面を構成する各GUI部品を描画する際には、それぞれのGUI部品クラスのメソッドが制御部4によって実行され、ポインタを用いてGUI部品インスタンス及びGUI部品画面テーマインスタンスを参照しながら、GUI画面を構成するGUI部品のそれぞれが描画され、結果的にGUI画面が描画されることになる。

【0040】

例えば、ボタンクラスのメソッドが制御部4によって実行されると、ポインタを用いてボタン1インスタンス及びボタン画面テーマインスタンスを読み出して

、ボタン1が描画されることになる。つまり、例えば、ボタン1は、ボタン1インスタンスの持っている属性（例えばボタンの描画位置とサイズ、文字列）と、ボタン画面テーマインスタンスの持っている属性（例えばボタンの背景色、フォントフェイス、枠線の有無）とを合わせたものとして描画されることになる。

【0041】

同様に、ボタンクラスのメソッドが制御部4によって実行されると、ポインタを用いてボタン2インスタンス及びボタン画面テーマインスタンスを読み出して、ボタン2が描画されることになる。つまり、例えば、ボタン2は、ボタン2インスタンスの持っている属性（例えばボタンの描画位置とサイズ、文字列）と、ボタン画面テーマインスタンスの持っている属性（例えばボタンの背景色、フォントフェイス、枠線の有無）とを合わせたものとして描画されることになる。

【0042】

ところで、画面テーマの切替時には、GUI画面クラス、各GUI部品クラス、GUI画面テーマクラス及び各GUI部品画面テーマクラスのメソッドが制御部4によって実行され、切替後の画面テーマのリソースファイル名を用いて画面テーマリソースファイルから画面テーマ定義データが読み出され、この画面テーマ定義データを用いて、アプリのメモリ領域に記憶されているGUI部品画面テーマインスタンスを更新するようになっている。

【0043】

つまり、本実施形態では、画面テーマの切替時には、アプリ起動時にGUI部品画面テーマクラスのコンストラクタによって予め初期化されているGUI部品画面テーマインスタンスの属性を画面テーマ定義データを用いて設定しなおすことで、GUI部品画面テーマインスタンスを更新するようにしている。なお、本実施形態では、アプリ起動時に生成したGUI部品画面テーマインスタンスはアプリケーションの終了時まで削除されない。

【0044】

このように、本実施形態では、画面テーマの切替時には、新たにGUI部品画面テーマインスタンスの生成を行わず、また、GUI部品画面テーマインスタンスの初期化処理（例えばnew演算子を用いる処理）も行なわないため、画面

テーマの高速切替を実現できることになる。また、アプリの初期化から終了までメモリの使用領域が変動しないため、アプリのために確保するメモリ領域を少なくすることができ、これにより、メモリ容量の少ない携帯電話において画面テーマの切替機能を実現できることになる。

【 0 0 4 5 】

このようにして、G U I 画面テーマクラスを含むものとして構成されるアプリケーションプログラムにしたがって、制御部 4 が、切替後の画面テーマの画面テーマ定義データを画面テーマリソースファイルから読み出し、アプリのために確保されたメモリ領域に記憶させる切替時画面テーマ読出処理ステップを実行することになる。この機能を切替時画面テーマ読出処理手段という。ここでは、画面テーマ定義データは、G U I 部品画面テーマインスタンスの属性値としてアプリのメモリ領域に記憶される。

【 0 0 4 6 】

そして、G U I 部品画面テーマインスタンスを更新すると、G U I 画面テーマクラスのメソッドが制御部 4 によって実行され、G U I 画面クラス（具体的には各 G U I 部品クラス）に対して画面無効化通知がなされ、これに応じて、（G U I 画面クラス（具体的には各 G U I 部品クラス）のメソッドが制御部 4 によって実行され、G U I 部品インスタンス及び G U I 部品画面テーマインスタンスを参照しながら、G U I 画面を構成する G U I 部品のそれぞれが再描画され、結果的に G U I 画面が再描画されることになる。

【 0 0 4 7 】

この場合、各 G U I 部品クラスのメソッドでは、自らのプロセス内のメモリアクセスで、G U I 部品インスタンスを参照できるだけでなく、さらに G U I 部品画面テーマインスタンスを参照して画面テーマ定義データを取得できることになる。これにより、画面テーマの高速切替が実現できることになる。

例えば、ボタンクラスのメソッドが制御部 4 によって実行されると、ポインタを用いて、ボタン 1 インスタンス及びボタン画面テーマインスタンスを読み出しながら、ボタン 1 が再描画されることになる。同様に、ボタンクラスのメソッドが制御部 4 によって実行されると、ポインタを用いて、ボタン 2 インスタンス及

びボタン画面テーマインスタンスを読み出しながら、ボタン 2 が再描画されることになる。

(3) また、本実施形態では、現在の画面テーマが、複数の画面テーマの中のどの画面テーマに設定されているかを、システム全体で 1 個のリソースファイル名(画面テーマに関する情報; 例えばポインタ)によって管理する。

【0048】

具体的には、現在の画面テーマリソースファイルのファイル名(以下、リソースファイル名という)を記憶部(例えばフラッシュメモリ)5に格納しておき、携帯電話1の電源投入時(OS起動時)に、OSのために確保されたメモリ領域にロードされるようにして(ここではカーネルのメモリ領域にカーネルオブジェクトとしてロードされるようにして)、OSによって現在の画面テーマリソースファイル名が管理されるようにしておく。これにより、OSによって管理される画面テーマリソースファイル名を用いて画面テーマリソースファイルから画面テーマ定義データを読み出すことが可能となる。

【0049】

このように、現在の画面テーマのリソースファイル名をシステム全体で1個だけ持つようにしているのは、以下の理由による。

つまり、各アプリケーションのメモリ領域のそれぞれに現在のリソースファイル名を持たせておくと、画面テーマの切替時に全てのアプリケーションが持っている現在のリソースファイル名を書き換えなくてはならず、アプリケーション間のコンテキストスイッチングの回数が多くなってしまい、オーバーヘッドが発生するため、画面テーマの切替が大変遅くなってしまう。

【0050】

これに対し、現在の画面テーマのリソースファイル名をシステム全体で1個だけ持つようにしておけば、1個のリソースファイル名を1回だけ書き換えるだけで良く、アプリケーション間のコンテキストスイッチングの回数を減らすことができ、オーバーヘッドを抑えることができるため、画面テーマの高速切替を実現できるようになる。

【0051】

そして、画面テーマの切替時には、ユーザによる画面テーマ決定指令に基づいて、OS（ここではカーネル）のメモリ領域に格納されている現在のリソースファイル名が書き換えられるようにしておき、これが書き換えられた場合に、OS（ここではカーネル）が、現在のリソースファイル名が書き換えられた時点で起動している各アプリ（ここではGUI画面テーマクラス）に対して画面テーマの変更通知を行ない、これに応じて、各アプリが再描画を行なうようにする。

【0052】

例えば、OSが、起動している各アプリに対してイベントを発生させるようにポインタを設定できるようになっているならば、このOSの機能を利用して、各アプリ（ここではGUI画面テーマクラス）に対してイベントを発生させるようにポインタを設定しておき、現在のリソースファイル名が書き換えられた場合に、OSが、ポインタを用いて各アプリ（ここではGUI画面テーマクラス）に対してイベントを発生させることで、画面テーマの変更通知が行なわれるようにすれば良い。

【0053】

なお、記憶部（例えばフラッシュメモリ）5には、最初、現在の画面テーマのリソースファイル名として、デフォルトとして設定された画面テーマのリソースファイル名が格納されているが、画面テーマが切り替えられ、カーネルのメモリ領域上のリソースファイル名が書き換えられると、記憶部5に格納されているリソースファイル名も書き換えられるようになっている。これにより、現在の画面テーマのリソースファイル名は、電源OFF後も保存されることになる。

【0054】

また、本実施形態の携帯電話1には、ユーザが任意にGUI画面の画面テーマを設定できるように、画面テーマを設定するのに用いる画面テーマ設定アプリ（スクリーン設定アプリ）が備えられている。この画面テーマ設定アプリケーションは、上述のアプリケーションと同様に構成され、GUI画面の画面テーマが高速で切り替わるようになっている。なお、本実施形態では、画面テーマ設定アプリのほかに画面テーママネージャも備えられているが、画面テーママネージャによる処理（機能）を全て画面テーマ設定アプリに含ませて構成しても良い。

【 0 0 5 5 】

本実施形態の携帯電話 1 は、上述のように構成されるため、以下、制御部 4 によって行なわれる画面テーマの切り替えにかかわる処理について、図 2 ～図 6 を参照しながら説明する。

①アプリ起動時の G U I 画面描画処理

まず、アプリ起動時に、上述のように G U I 画面クラスや G U I 画面テーマクラスを含むアプリケーションプログラムにしたがって制御部（コンピュータ）4 が実行する G U I 画面の描画処理について、図 2 を参照しながら説明する。

【 0 0 5 6 】

なお、ここでは、G U I 画面の構成要素としてのボタン（ここでは 2 つのボタン 1，2）の描画処理について説明するが、例えばリストなどの他の構成要素についても同様の描画処理が行なわれる。

図 2 に示すように、アプリケーションの G U I 画面クラスは、G U I 画面を構成するボタン 1 を描画すべく、ボタンクラスに対して、ボタン 1 の描画を要求するメッセージを出す（G U I 画面クラスのコンストラクタによる処理）。

【 0 0 5 7 】

これに応じて、ボタンクラスは、ボタン画面テーマクラスに対して、ボタンの画面テーマを要求するメッセージを出す（ボタンクラスのコンストラクタによる処理）。なお、ボタンクラスのコンストラクタは、ボタンクラスによって生成されるインスタンス（オブジェクト）の初期化を行なう。

ボタン画面テーマクラスは、ボタンクラスからのメッセージを受けると、コンストラクタによって、以下のようにして、ボタン画面テーマクラスのインスタンス（ボタン画面テーマインスタンス）を生成する。なお、ボタン画面テーマクラスのコンストラクタは、ボタン画面テーマクラスによって生成されるインスタンスの初期化を行なう。

【 0 0 5 8 】

まず、ボタン画面テーマクラスは、O S（ここではカーネル）に対して、画面テーマリソースファイルのファイル名（以下、リソースファイル名という）を要求するメッセージを出す。

これに応じて、OS（ここではカーネル）は、OS（ここではカーネル）のために確保されたメモリ領域（OS用メモリ領域、カーネル用メモリ領域）に記憶されて管理される現在の画面テーマのリソースファイル名（画面テーマに関する情報；例えばポインタ）を読み出して、これをボタン画面テーマクラスに通知する。

【0059】

なお、本実施形態では、複数の画面テーマリソースファイルを備えるが、OS起動時（電源投入時）には、デフォルトとして設定された画面テーマのリソースファイル名が、現在のリソースファイル名として、OS（ここではカーネル）のメモリ領域に記憶されるようになっている。

ボタン画面テーマクラスは、リソースファイル名を受け取ると、OSに対してボタンの画面テーマ定義データを要求する。ここでは、ボタン画面テーマクラスは、リソースファイル名及びボタンの画面テーマ定義データが格納されているアドレスをOSに対して通知して、ボタンの画面テーマ定義データを要求する。

【0060】

これに対して、OSは、リソースファイル名及びボタンの画面テーマ定義データが格納されているアドレスを用いて、該当する画面テーマリソースファイルからボタンの画面テーマ定義データを読み出し、これをボタン画面テーマクラスに対して通知する。

このように、本実施形態では、制御部4は、アプリケーションプログラムにしたがって、OSによって管理される現在の画面テーマのリソースファイル名を用いて画面テーマリソースファイルから画面テーマ定義データを読み出す処理を実行するようになっている。

【0061】

次に、ボタン画面テーマクラスは、ボタンの画面テーマ定義データを受け取ると、これに基づいてボタン画面テーマインスタンス（オブジェクト）を生成し、アプリケーションプログラムのために確保されたメモリ領域（アプリ用メモリ領域）上に領域（ボタン画面テーマインスタンス用メモリ領域）を確保して、生成したボタン画面テーマインスタンスを格納する。また、ボタン画面テーマクラス

は、ボタン画面テーマインスタンスを格納したメモリ領域のアドレスを記憶する
[ここではポインタ（以下、ボタン画面テーマ用ポインタという）を設定する]
（ステップ S 1 0）。

【 0 0 6 2 】

このように、ボタン画面テーマクラス（G U I 画面テーマクラス）を含むアプリにしたがって、制御部 4 によって、アプリ起動時に、デフォルトとして設定された画面テーマの画面テーマ定義データを画面テーマリソースファイルから読み出し、アプリのメモリ領域に記憶させる処理（アプリ起動時画面テーマ読出処理ステップ）が実行される。この機能をアプリ起動時画面テーマ読出処理手段という。

【 0 0 6 3 】

そして、ボタン画面テーマクラスは、上述のボタンの画面テーマを要求するメッセージに対する応答として、ボタン画面テーマ用ポインタをボタンクラスに対して返す。

ボタンクラスは、ボタン画面テーマ用ポインタを受け取ると、これを記憶する。そして、ボタンクラスは、ボタン 1 のインスタンスを生成し、アプリ用メモリ領域上に領域（ボタン 1 インスタンス用メモリ領域）を確保して、生成したボタン 1 のインスタンスを格納する。また、ボタンクラスは、ボタン 1 のインスタンスを格納したメモリ領域のアドレスを記憶する [ここではポインタ（以下、ボタン 1 用ポインタという）を設定する]（ステップ S 2 0）。

【 0 0 6 4 】

このようにして、ボタン画面テーマクラスがボタン画面テーマインスタンスを生成し、ボタンクラスがボタン 1 のインスタンスを生成した後、ボタンクラスは、これらのインスタンスを用いてボタン 1 を描画する（ステップ S 3 0）。つまり、ボタンクラスは、ボタン 1 用ポインタを用いてボタン 1 のインスタンスを読み出し、さらにボタン画面テーマ用ポインタを用いてボタン画面テーマインスタンスを読み出して、ボタン 1 を描画する。

【 0 0 6 5 】

そして、ボタン 1 の描画が終了すると、ボタンクラスは、G U I 画面クラスに

対してボタン 1 描画完了通知を行なう。

次いで、GUI 画面クラスは、GUI 画面を構成するボタン 2 を描画すべく、図 2 に示すように、ボタンクラスに対して、ボタン 2 の描画を要求するメッセージを出す（アプリ画面クラスのコンストラクタによる処理）。これに応じて、ボタンクラスは、ボタン画面テーマクラスに対して、ボタンの画面テーマを要求するメッセージを出す（ボタンクラスのコンストラクタによる処理）。

【0066】

ボタン画面テーマクラスは、このメッセージを受けると、ボタン画面テーマ用ポインタをボタンクラスに対して通知する。

ボタンクラスは、ボタン画面テーマ用ポインタを受け取ると、これを記憶する。そして、ボタン 2 のインスタンスを生成し、アプリ用メモリ領域上に領域（ボタン 2 インスタンス用メモリ領域）を確保して、生成したボタン 2 のインスタンスを格納する。そして、ボタンクラスは、ボタン 2 のインスタンスを格納したメモリ領域のアドレスを記憶する〔ここではポインタ（以下、ボタン 2 用ポインタという）を設定する〕（ステップ S 4 0）。

【0067】

このようにして、ボタン 2 のインスタンスを生成した後、ボタンクラスは、このボタン 2 のインスタンス及び上述のボタン画面テーマインスタンスを用いてボタン 2 を描画する。つまり、ボタンクラスは、ボタン 2 用ポインタを用いてボタン 2 のインスタンスを読み出し、さらにボタン画面テーマ用ポインタを用いてボタン画面テーマインスタンスを読み出して、ボタン 2 を描画する（ステップ S 5 0）。

【0068】

そして、ボタン 2 の描画が終了すると、ボタンクラスは、アプリ画面クラスに対してボタン 2 描画完了通知を行なう。

以後、GUI 画面に含まれるボタンの数に応じて、上述のボタン 2 を描画する場合の処理が繰り返される。

②画面テーマ設定アプリケーションを用いた画面テーマ設定処理

次に、画面テーマ設定アプリ（ここでは画面テーママネージャも含む）にした

がって制御部（コンピュータ）4 が実行する画面テーマの設定処理について、図 3，図 4 を参照しながら説明する。

【 0 0 6 9 】

ここでは、最初に、画面テーマ設定アプリ（ここでは画面テーママネージャも含む）にしたがって制御部 4 が実行する画面テーマ設定アプリの G U I 画面の描画処理について、図 3 を参照しながら説明し、次いで、画面テーマ設定アプリの G U I 画面を介して画面テーマの設定を行なう場合に、画面テーマ設定アプリ（ここでは画面テーママネージャも含む）にしたがって制御部 4 が実行する画面テーマ設定処理について、図 4 を参照しながら説明する。

【 0 0 7 0 】

なお、画面テーマ設定アプリによる G U I 画面の描画処理については、基本的には、上述したアプリによる G U I 画面の描画処理（図 2 参照）と同様である。

最初に、図 3 に示すように、ユーザが、画面テーマを設定すべく、画面テーマ設定アプリを起動させると、画面テーマリソースファイルのファイル数に応じた数の欄を有するリストを作成すべく、画面テーマ設定アプリ及び画面テーママネージャにしたがって、制御部 4 によって、以下のリスト作成処理が行なわれる。

【 0 0 7 1 】

まず、画面テーマ設定アプリは、画面テーママネージャに対して画面テーマリソースファイルのファイル数（以下、リソースファイル数という）を要求する。

画面テーママネージャは、O S に対して画面テーマリソースファイルのファイル名（以下、リソースファイル名という）を要求する。これに応じて、O S は、ファイルシステムの画面テーマリソースファイル用ディレクトリに格納されている全ての画面テーマリソースのファイル名を取得し、これを画面テーママネージャに対して返答する。

【 0 0 7 2 】

画面テーママネージャは、リソースファイル名を取得すると、それぞれのリソースファイル名にテーマ番号（テーマ N O . ）を付け、テーマ N O . とリソースファイル名とを対応づけた対応表を作成する（ステップ A 1 0）。また、画面テーママネージャは、取得されたリソースファイル名に基づいてリソースファイル

数を求め、画面テーマ設定アプリに返答する。

画面テーマ設定アプリは、リソースファイル数を受け取ると、リソースファイル数に応じた数の欄を有するリストを作成する（ステップA20）。なお、リストの各欄は、画面テーママネージャが管理するテーマNO. とリソースファイル名との対応表のテーマNO. と関連づけられている。

【0073】

次に、上述のようにして作成されるリストの各欄にタイトル文字列を組み込むべく、画面テーマ設定アプリ（ここでは画面テーママネージャも含む）にしたがって、制御部4によって、以下のタイトル文字列組込処理が行なわれる。

まず、図3に示すように、画面テーマ設定アプリは、画面テーママネージャに対して、リストの各欄に表示すべき画面テーマのタイトル文字列を各欄毎に要求する。

【0074】

これに応じて、画面テーママネージャは、要求された欄に関連づけられているテーマNO. に基づいてリソースファイル名を特定し（ステップA30）、このリソースファイル名に基づいて、OSに対して、画面テーマリソースファイルに格納されている画面テーマのタイトル文字列を要求する。

そして、画面テーママネージャは、OSからタイトル文字列の返答があると、これを要求された欄に表示すべき画面テーマのタイトル文字列として、画面テーマ設定アプリに返答する。

【0075】

画面テーマ設定アプリは、画面テーマのタイトル文字列の返答があると、上述のようにして作成したリストの各欄に、対応する画面テーマのタイトル文字列を組み込む処理（タイトル文字列組込処理）を行なう（ステップA40）。

次いで、現在の画面テーマのタイトル文字列が表示される欄と他の画面テーマのタイトル文字列が表示される欄とをユーザが識別できるように、現在の画面テーマのタイトル文字列が表示される欄にフォーカスを当てるべく（強調表示すべく）、画面テーマ設定アプリ（ここでは画面テーママネージャも含む）にしたがって、制御部4によって、以下の処理が行なわれる。

【 0 0 7 6 】

まず、図 3 に示すように、画面テーマ設定アプリは、画面テーママネージャに対して、現在の画面テーマのテーマ NO. を要求する。

これに応じて、画面テーママネージャは、OS（ここではカーネル）に対して、現在の画面テーマのリソースファイル名を要求する。

そして、画面テーママネージャは、OS（ここではカーネル）から現在の画面テーマのリソースファイル名の返答があると、テーマ NO. とリソースファイル名とを対応づけて対応表を用いて、現在の画面テーマのリソースファイル名に対応するテーマ NO. を特定し（ステップ A 5 0）、これを現在の画面テーマのテーマ NO. として、画面テーマ設定アプリに返答する。

【 0 0 7 7 】

画面テーマ設定アプリは、現在の画面テーマのテーマ NO. の返答を受けると、この現在の画面テーマのテーマ NO. に関連づけられている欄を現在の画面テーマのタイトル文字列を表示する欄として特定し、この欄と他の画面テーマのタイトル文字列が表示される欄とをユーザが識別できるように、現在の画面テーマのタイトル文字列を表示する欄にフォーカスが当たるようにして画面テーマリスト（画面テーマ一覧）を完成させ、この画面テーマリストを含む GUI 画面をディスプレイ上に表示させる（ステップ A 6 0）。

【 0 0 7 8 】

ここで、図 7 では、画面テーマ設定アプリの GUI 画面として、8 つの欄を有する画面テーマリスト、選択ボタン、スクロールボタン、アンテナ・電池残量等を含む GUI 画面をディスプレイ上に表示した状態を例示している。

また、図 7 中、画面テーマリストの第 1 欄から第 8 欄までの各欄に順に、“AccessoriesRuby”、“AccessoriesSapphire”、“ClassicalGreen”、“ClassicalTomato”、“FancyOrange”、“FancyPink”、“ForestAqua”、“ForestYellow”と組み込まれている文字列が、画面テーマのタイトル文字列である。

【 0 0 7 9 】

さらに、図 7 では、現在の画面テーマのタイトル文字列を表示する欄として、ユーザが識別しうるように、“FancyPink”というタイトル文字列が表示されてい

る第 1 欄にフォーカスが当てられた状態を例示している。

また、現在の画面テーマが“FancyPink”という画面テーマになっており、上述の画面テーマ設定アプリによる G U I 画面の描画処理（例えばステップ A 2 0，A 4 0，A 6 0）として、上述したアプリによる G U I 画面の描画処理（図 2 参照）と同様の処理が行なわれるため、例えば図 7 に示すように、画面テーマ設定アプリの G U I 画面が、“FancyPink”という画面テーマで全体的に統一されたものとして表示される。

【 0 0 8 0 】

次に、上述のようにしてディスプレイ上に表示された画面テーマ設定アプリの G U I 画面を介して画面テーマの設定を行なう場合に、画面テーマ設定アプリ（ここでは画面テーママネージャも含む）にしたがって制御部 4 が実行する画面テーマ設定処理（画面テーマを他の画面テーマに変更する処理）について、図 4 を参照しながら説明する。

【 0 0 8 1 】

まず、ユーザが操作部 3 を操作して他の画面テーマのタイトル文字列が表示されている欄を選択すると、画面テーマ設定アプリは、画面テーママネージャに対して、選択された画面テーマの画面テーマ定義データを要求する。

これに応じて、画面テーママネージャは、テーマ N O . とリソースファイル名とを対応づけて対応表を用いて、選択された画面テーマのタイトル文字列が表示されている欄に関連づけられているテーマ N O . に対応するリソースファイル名を特定し（ステップ B 1 0）、このリソースファイル名に基づいて、O S に対して、画面テーマ定義データを要求する。

【 0 0 8 2 】

そして、画面テーママネージャは、O S から画面テーマ定義データの返答を受けると、これを選択された画面テーマ定義データとして、画面テーマ設定アプリに返答する。

画面テーマ設定アプリは、選択された画面テーマ定義データの返答を受けると、画面テーマのリストを含む G U I 画面の画面テーマを切り替えて表示する（ステップ B 2 0）。具体的には、画面テーマ設定アプリは、後述する G U I 画面の

再描画処理（図 6 参照）と同様の処理を行なって、画面テーマ設定アプリの G U I 画面を再描画することで、G U I 画面の画面テーマを切り替えることになる。

【 0 0 8 3 】

このように、制御部 4 が、画面テーマ設定アプリ（ここでは画面テーママネージャも含む）にしたがって、画面テーマ設定アプリの G U I 画面に含まれる画面テーマリストの選択位置に応じて、G U I 画面の画面テーマを切り替える処理を実行するようにしている。

その後、ユーザが、操作部 3 を操作して図 7 の G U I 画面上に表示されている「選択」ボタンを押すことで、画面テーマを他の画面テーマに切り替えることを決定する指令（画面テーマ決定指令）を出した場合、画面テーマ設定アプリは、画面テーママネージャに対して、決定された画面テーマのタイトル文字列が表示されている欄のテーマ N O . を通知して、現在の画面テーマのリソースファイル名を書き換えるように要求する。

【 0 0 8 4 】

これに応じて、画面テーママネージャは、テーマ N O . とリソースファイル名とを対応づけて対応表を用いて、決定された画面テーマのタイトル文字列が表示されている欄に関連づけられているテーマ N O . に対応するリソースファイル名を特定し（ステップ B 3 0 ）、O S （ここではカーネル）に対して、特定されたリソースファイル名を通知して、現在の画面テーマのリソースファイル名を書き換えるように要求する。

【 0 0 8 5 】

O S （ここではカーネル）は、画面テーママネージャからの現在の画面テーマのリソースファイル名の書換要求を受けると、自らが管理している現在の画面テーマのリソースファイル名を、特定されたリソースファイル名に書き換える（ステップ B 4 0 ）。

なお、ここでは、ユーザによる画面テーマの選択時に、画面テーマ設定アプリの G U I 画面自体の画面テーマを切り替えるようにしているが、これに限られるものではなく、例えば、予め画面テーマリソースファイルにそれぞれの画面テーマに対応するビットマップイメージを格納しておき、制御部 4 が、画面テーマ設

定アプリ（ここでは画面テーママネージャも含む）にしたがって、以下のような処理を行なうことで、画面テーマ設定アプリのGUI画面上にビットマップイメージを表示させるようにしても良い。

【0086】

この場合、まず、図5に示すように、ユーザが操作部を操作して他の画面テーマのタイトル文字列が表示されている欄を選択すると、画面テーマ設定アプリは、画面テーママネージャに対して、選択された画面テーマのビットマップイメージを要求する。

これに応じて、画面テーママネージャは、テーマNO. とリソースファイル名とを対応づけて対応表を用いて、選択された画面テーマのタイトル文字列が表示されている欄に関連づけられているテーマNO. に対応するリソースファイル名を特定し（ステップC10）、このリソースファイル名に基づいて、OSに対して、画面テーマのビットマップイメージを要求する。

【0087】

そして、画面テーママネージャは、OSから画面テーマのビットマップイメージの返答を受けると、これを選択された画面テーマのビットマップイメージとして、画面テーマ設定アプリに返答する。

画面テーマ設定アプリは、選択された画面テーマのビットマップイメージの返答を受けると、画面テーマリストを含むGUI画面上にビットマップイメージを表示させる（ステップC20）。

【0088】

このように、制御部4が、画面テーマ設定アプリ及び画面テーママネージャにしたがって、画面テーマ設定アプリのGUI画面に含まれる画面テーマリストの選択位置に応じて、GUI画面上にビットマップイメージを表示させる処理を実行するようにしている。

その後、ユーザが、操作部を操作して図7のGUI画面上に表示されている「選択」ボタンを押すことで、画面テーマを他の画面テーマに切り替えることを決定する指令（画面テーマ決定指令）を出した場合、画面テーマ設定アプリは、画面テーママネージャに対して、現在の画面テーマのリソースファイル名の変更を

要求する。

【0089】

これに応じて、画面テーママネージャは、テーマNO. とリソースファイル名とを対応づけて対応表を用いて、決定された画面テーマのタイトル文字列が表示されている欄に関連づけられているテーマNO. に対応するリソースファイル名を特定し（ステップC30）、OS（ここではカーネル）に対して、特定されたリソースファイル名を通知して、現在の画面テーマのリソースファイル名を書き換えるように要求する。

【0090】

OS（ここではカーネル）は、画面テーママネージャからの現在の画面テーマのリソースファイル名の書換要求を受けると、自らが管理している現在の画面テーマのリソースファイル名を、特定されたリソースファイル名に書き換える（ステップC40）。

③画面テーマ切替時のアプリのGUI画面の再描画処理

次に、上述のようにして画面テーマを切り替えるべく他の画面テーマの設定が行なわれた場合に、GUI画面クラス及びGUI画面テーマクラスを含むアプリにしたがって、制御部4が実行するアプリのGUI画面の再描画処理（画面テーマ切替処理）について、図6を参照しながら説明する。

【0091】

なお、ここでは、GUI画面の構成要素としてのボタン（ここでは2つのボタン1，2）の再描画処理について説明するが、例えばリストなどの他の構成要素についても同様の再描画処理が行なわれる。

まず、上述のようにして画面テーマを他の画面テーマに切り替える決定がなされ、OS（ここではカーネル）のメモリ領域に記憶されている現在のリソースファイル名（画面テーマに関する情報）が書き換えられた場合に、図6に示すように、OS（ここではカーネル）が、現在のリソースファイル名が書き換えられた時点で起動している各アプリ（ここではGUI画面テーマクラス）に対して、画面テーマの変更通知がなされる。

【0092】

本実施形態では、OSが、起動している各アプリケーションに対してイベントを発することができるようにポインタの設定が可能となっているため、このOSの機能を利用して、各アプリ（ここではGUI画面テーマクラス）に対してイベントを発することができるようにポインタを設定しておき、現在のリソースファイル名が書き換えられた場合に、OSが、ポインタを用いて各アプリ（ここではGUI画面テーマクラス）に対してイベントを発することで、画面テーマの変更通知が行なわれるようになっている。

【0093】

GUI画面テーマクラスは、画面テーマの変更通知（イベント）を受けると、OS（ここではカーネル）に対して、現在の画面テーマのリソースファイル名を要求するメッセージを出す。

具体的には、GUI画面テーマクラスは、例えばボタン画面テーマクラスなどのGUI部品画面テーマクラスを集約したものとして構成されるため、各GUI部品画面テーマクラスが、OS（ここではカーネル）に対して、現在の画面テーマのリソースファイル名を要求するメッセージを出す。

【0094】

これに応じて、OS（ここではカーネル）は、OS（ここではカーネル）のメモリ領域に記憶されている現在のリソースファイル名を読み出して、これをGUI画面テーマクラス（具体的には各GUI部品画面テーマクラス）に通知する。なお、この段階では、現在のリソースファイル名は画面テーマ切替後のリソースファイル名に書き換えられている。

【0095】

GUI画面テーマクラス（具体的にはGUI部品画面テーマクラス）は、リソースファイル名を受け取ると、OSに対して、それぞれのGUI部品（例えばボタン）の画面テーマ定義データを要求する。ここでは、GUI画面テーマクラス（具体的にはGUI部品画面テーマクラス）は、リソースファイル名及びそれぞれのGUI部品（例えばボタン）の画面テーマ定義データが格納されているアドレスをOSに対して通知して、それぞれのGUI部品（例えばボタン）の画面テーマ定義データを要求する。

【 0 0 9 6 】

これに対して、OSは、リソースファイル名及びそれぞれのGUI部品（例えばボタン）の画面テーマ定義データが格納されているアドレスを用いて、該当する画面テーマリソースファイルからそれぞれのGUI部品（例えばボタン）の画面テーマ定義データを読み出し、これをGUI画面テーマクラス（具体的にはGUI部品画面テーマクラス）に通知する。

【 0 0 9 7 】

GUI画面テーマクラス（具体的にはGUI部品画面テーマクラス）は、各GUI部品（例えばボタン）の画面テーマ定義データを受け取ると、これらを用いて、GUI部品画面テーマインスタンス用メモリ領域（例えばボタン画面テーマインスタンス用メモリ領域）に格納されているそれぞれのGUI部品画面テーマインスタンス（例えばボタン画面テーマインスタンス）を更新する（ステップD10）。

【 0 0 9 8 】

つまり、アプリ起動時に、GUI部品画面テーマクラス（例えばボタン画面テーマインスタンス）のコンストラクタによって予め初期化されているGUI部品画面テーマインスタンス（例えばボタン画面テーマインスタンス）の属性を、画面テーマ定義データを用いて設定しなおすことでGUI部品画面テーマインスタンスを更新する。

【 0 0 9 9 】

このようにして、GUI画面テーマクラスを含むものとして構成されるアプリケーションプログラムにしたがって、制御部4が、切替後の画面テーマの画面テーマ定義データを画面テーマリソースファイルから読み出し、アプリのために確保されたメモリ領域に記憶させる切替時画面テーマ読出処理ステップを実行することになる。ここでは、画面テーマ定義データは、GUI部品画面テーマインスタンスの属性値としてアプリのメモリ領域に記憶される。

【 0 1 0 0 】

この場合、画面テーマ切替後のGUI部品画面テーマインスタンス（例えばボタン画面テーマインスタンス）が格納されているGUI部品画面テーマインスタ

ンス用メモリ領域のアドレスは、画面テーマ切替前のアドレスと変わらないため、G U I 部品画面テーマ用ポインタ（例えばボタン画面テーマ用ポインタ）は変更されない。

【0 1 0 1】

このようにして、G U I 画面テーマクラス（具体的にはG U I 部品画面テーマクラス）が、G U I 画面を構成するG U I 部品の画面テーマインスタンス（G U I 部品画面テーマインスタンス）の全てを更新した後、G U I 画面クラスに対してG U I 画面の無効化通知（画面無効化通知）を行なう。

G U I 画面クラスが、画面無効化通知を受けると、G U I 画面クラスに含まれる各G U I 部品クラスがそれぞれ各G U I 部品を再描画する（ステップD 2 0）。

【0 1 0 2】

例えば、ボタン 1，ボタン 2 を含むG U I 画面を再描画する場合を例に説明する。

まず、G U I 画面クラスは、ボタンクラスに対して、ボタン 1 の再描画を要求するメッセージを出す。これに応じて、ボタンクラスは、更新されたボタン画面テーマインスタンス及びボタン 1 インスタンスを用いてボタン 1 を再描画する。つまり、ボタンクラスは、ボタン 1 インスタンス用ポインタを用いてボタン 1 インスタンスを読み出し、さらにボタン画面テーマインスタンス用ポインタを用いてボタン画面テーマインスタンスを読み出して、ボタン 1 を再描画する。

【0 1 0 3】

次いで、G U I 画面クラスは、ボタンクラスに対して、ボタン 2 の再描画を要求するメッセージを出す。これに応じて、ボタンクラスは、更新されたボタン画面テーマインスタンス及びボタン 2 インスタンスを用いてボタン 2 を再描画する。つまり、ボタンクラスは、ボタン 2 インスタンス用ポインタを用いてボタン 2 インスタンスを読み出し、さらにボタン画面テーマインスタンス用ポインタを用いてボタン画面テーマインスタンスを読み出して、ボタン 2 を再描画する。

【0 1 0 4】

このように、ボタンクラスは、切替後の画面テーマの画面テーマ定義データを

OSによって管理されている画面テーマリソースファイルから読み出すのではなく、上述のようにしてアプリのメモリ領域に記憶した画面テーマ定義データ（ボタン画面テーマインスタンス）を用いて、2つのボタン1，ボタン2を再描画することになる。

【0105】

これは、GUI画面クラスを含むものとして構成されるアプリケーションプログラムにしたがって、制御部4が、アプリのメモリ領域に記憶されている画面テーマ定義データ（GUI部品画面テーマインスタンス）を用いてGUI画面に含まれる複数のGUI部品を描画するGUI画面描画処理ステップを実行することを意味する。この機能をGUI画面描画処理手段という。

【0106】

このようにして、アプリとOSとの間のコンテキストスイッチングを減らして、オーバーヘッドを抑えるようにして、各アプリの画面テーマの切替が高速で行なわれるようにしている。

例えば画面テーマが“FancyPink”に設定されている場合（図7参照）、各アプリのGUI画面（ここでは電話帳アプリのGUI画面）は、例えば図8に示すような画面として表示されるが、これを、上述のようにして画面テーマ設定アプリを用いて画面テーマを切り替えて、切替後の画面テーマを例えば“ForestYellow”に設定した場合（図9参照）、各アプリのGUI画面（ここでは電話帳アプリのGUI画面）は、例えば図10に示すような画面に高速で切り替わることになる。

【0107】

したがって、本実施形態にかかる携帯端末としての携帯電話及びアプリケーションプログラムによれば、アプリケーションの画面が全体的に統一されたテーマ（画面テーマ）で表現され、このような画面テーマを高速で切り替える機能を実装できるという利点がある。また、できるだけ少ないメモリ領域を使用しながら、画面テーマの切替（特に高速切替）を実現できるという利点もある。

【0108】

なお、上述の実施形態では、携帯電話を例に説明したが、携帯端末は、携帯電

話に限られるものではなく、例えば携帯情報端末（PDA；Personal Digital Assistants）などの携帯型の端末が含まれる。また、上述の実施形態では、本発明を携帯電話に適用したものとして説明したが、本発明は、携帯電話をはじめとする携帯端末に適用できるだけでなく、画面テーマの切替機能を備える例えばパーソナルコンピュータのようなものにも適用できる。

【0109】

また、上述の実施形態では、画面テーマ切替処理を制御部4に実行させるアプリケーションプログラム（携帯電話用アプリケーションプログラム）を備える携帯電話について説明したが、このようなアプリケーションプログラムは、コンピュータ読取可能な記録媒体に格納することができ、記録媒体に格納した状態で頒布し、市場を流通させ、また売買することができる。

【0110】

そして、このような記録媒体に格納されているアプリケーションプログラムを携帯電話などの携帯端末やパーソナルコンピュータなどのコンピュータにインストールし、携帯端末やコンピュータがこのプログラムを実行することで、上述の実施形態で説明したものと同様の作用、効果が得られる。

ここで、記録媒体には、例えば半導体メモリなどのメモリ、磁気ディスク、光ディスク（例えばCD-ROM等）、光磁気ディスク（MO）、磁気テープ、ハードディスク、フレキシブルディスク、ICカード、ROMカートリッジ、パンチカード、コンピュータ内部の記憶装置（RAMやROMなどのメモリ）、外部記憶装置等のプログラムを記録することができるものが含まれる。また、バーコードなどの符号が印刷された印刷物等のコンピュータ読取可能な種々の媒体を利用することもできる。なお、CD-ROM、MO、磁気テープ、ICカード等を可搬型記録媒体ともいう。

【0111】

なお、ここでは、アプリケーションプログラムをコンピュータ読取可能な記録媒体に格納しているが、記録媒体に格納しなくても良い。また、アプリケーションプログラムは、上述のような記録媒体に格納せずに、例えば伝送媒体としてのネットワーク（通信ネットワーク）を介して伝送（送受信）することもでき、こ

れによりアプリケーションプログラムを頒布し、市場を流通させ、また売買することもできる。例えば、アプリケーションプログラムをウェブサーバ等にアップロードしたり、インターネット等のネットワークを経由して例えばウェブサーバ等からダウンロードしたりすることもできる。

【 0 1 1 2 】

また、本発明は、上述した実施形態に限定されず、上記以外にも、本発明の趣旨を逸脱しない範囲で種々変形して実施することができる。

(付記 1) 同種のグラフィカル・ユーザ・インターフェース（以下、G U I という）部品を複数個含む G U I 画面を表示させるアプリケーションプログラムの G U I 画面の画面テーマを切り替える処理を実行する制御部と、

画面テーマを定義する画面テーマ定義データを格納する記憶部とを備え、
前記制御部が、

切替後の画面テーマの画面テーマ定義データを前記記憶部から読み出し、前記アプリケーションプログラムのために確保されたメモリ領域に記憶させる切替時画面テーマ読出処理ステップと、

前記アプリケーションプログラムのメモリ領域に記憶されている画面テーマ定義データを用いて前記 G U I 画面に含まれる同種の複数個の G U I 部品を描画する G U I 画面描画処理ステップとを含む処理を実行するように構成されることを特徴とする、携帯端末。

【 0 1 1 3 】

(付記 2) 前記制御部が、前記切替時画面テーマ読出処理ステップにおいて、前記記憶部から前記 G U I 画面に含まれる G U I 部品の画面テーマ定義データのみを読み出し、前記アプリケーションプログラムのメモリ領域に記憶させる処理を実行するように構成されることを特徴とする、付記 1 記載の携帯端末。

(付記 3) 前記制御部が、

前記アプリケーションプログラムの起動時に、デフォルトとして設定された画面テーマの画面テーマ定義データを前記記憶部から読み出し、前記アプリケーションプログラムのメモリ領域に記憶させるアプリ起動時画面テーマ読出処理ステップを実行するように構成され、

前記切替時画面テーマ読出処理ステップにおいて、切替後の画面テーマの画面テーマ定義データを前記記憶部から読み出し、前記アプリケーションプログラムのメモリ領域に記憶されている画面テーマ定義データを更新する処理を実行するように構成されることを特徴とする、付記 1 又は 2 記載の携帯端末。

【 0 1 1 4 】

(付記 4) 前記制御部が、前記切替時画面テーマ読出処理ステップにおいて、オペレーティングシステムによって管理される画面テーマに関する情報を用いて前記記憶部から画面テーマ定義データを読み出す処理を実行するように構成されることを特徴とする、付記 1 ～ 3 のいずれか 1 項に記載の携帯端末。

(付記 5) 前記制御部が、

前記記憶部に格納されている画面テーマ定義データを読み出し、オペレーティングシステムのために確保されたメモリ領域に展開する処理を実行するように構成され、

前記記憶部から前記画面テーマ定義データを読み出すのに代えて、前記記憶部から読み出され、前記オペレーティングシステムのメモリ領域に展開されている前記画面テーマ定義データを読み出す処理を実行するように構成されることを特徴とする、付記 1 ～ 4 のいずれか 1 項に記載の携帯端末。

【 0 1 1 5 】

(付記 6) 前記制御部が、オペレーティングシステムによって管理される画面テーマに関する情報が書き換えられた場合に、前記切替時画面テーマ読出処理ステップ及び前記 G U I 画面描画処理ステップを実行するように構成されることを特徴とする、付記 1 ～ 5 のいずれか 1 項に記載の携帯端末。

(付記 7) 前記制御部が、

画面テーマの設定に用いる画面テーマ設定アプリケーションプログラムにしたがって、画面テーマ決定指令に応じて、オペレーティングシステムによって管理される画面テーマに関する情報を切替後の画面テーマに関する情報に書き換える処理を実行するように構成され、

前記オペレーティングシステムにしたがって、前記画面テーマに関する情報が書き換えられた場合に、前記アプリケーションプログラムに対して画面テーマ変

更通知を行なう処理を実行するように構成され、

前記アプリケーションプログラムにしたがって、前記オペレーティングシステムからの画面テーマ変更通知を受けた場合に、前記切替時画面テーマ読出処理ステップ及び前記GUI画面描画処理ステップを実行するように構成されることを特徴とする、付記1～5のいずれか1項に記載の携帯端末。

【0116】

(付記8) 前記制御部が、画面テーマの設定に用いる画面テーマ設定アプリケーションプログラムのGUI画面に含まれる画面テーマリストの選択位置に応じて、前記画面テーマ設定アプリケーションプログラムのGUI画面の画面テーマを切り替える処理を実行するように構成されることを特徴とする、付記1～7のいずれか1項に記載の携帯端末。

【0117】

(付記9) 前記制御部が、画面テーマの設定に用いる画面テーマ設定アプリケーションプログラムのGUI画面に含まれる画面テーマリストの選択位置に応じて、前記画面テーマ設定アプリケーションプログラムのGUI画面上にビットマップイメージを表示させる処理を実行するように構成されることを特徴とする、付記1～7のいずれか1項に記載の携帯端末。

【0118】

(付記10) 同種のグラフィカル・ユーザ・インターフェース(以下、GUIという)部品を複数個含むGUI画面の画面テーマを切り替える画面テーマ切替処理をコンピュータに実行させるためのアプリケーションプログラムであって、

前記画面テーマ切替処理が、

切替後の画面テーマの画面テーマ定義データを画面テーマリソースファイルから読み出し、前記アプリケーションプログラムのために確保されたメモリ領域に記憶させる切替時画面テーマ読出処理ステップと、

前記アプリケーションプログラムのメモリ領域に記憶されている画面テーマ定義データを用いて前記GUI画面に含まれる同種の複数個のGUI部品を描画するGUI画面描画処理ステップとを含むことを特徴とする、アプリケーションプ

ログラム。

【0 1 1 9】

（付記 1 1） 前記切替時画面テーマ読出処理ステップにおいて、前記画面テーマリソースファイルから前記 G U I 画面に含まれる G U I 部品の画面テーマ定義データのみを読み出し、前記アプリケーションプログラムのメモリ領域に記憶させる処理を前記コンピュータに実行させるように構成されることを特徴とする、付記 1 0 記載のアプリケーションプログラム。

【0 1 2 0】

（付記 1 2） 起動時に、デフォルトとして設定された画面テーマの画面テーマ定義データを前記画面テーマリソースファイルから読み出し、前記アプリケーションプログラムのメモリ領域に記憶させる起動時画面テーマ読出処理ステップを前記コンピュータに実行させるように構成され、

前記切替時画面テーマ読出処理ステップにおいて、切替後の画面テーマの画面テーマ定義データを前記画面テーマリソースファイルから読み出し、前記アプリケーションプログラムのメモリ領域に記憶されている画面テーマ定義データを更新する処理を前記コンピュータに実行させるように構成されることを特徴とする、付記 1 0 又は 1 1 記載のアプリケーションプログラム。

【0 1 2 1】

（付記 1 3） 前記切替時画面テーマ読出処理ステップにおいて、オペレーティングシステムによって管理される画面テーマに関する情報を用いて前記画面テーマリソースファイルから画面テーマ定義データを読み出す処理を前記コンピュータに実行させるように構成されることを特徴とする、付記 1 0 ～ 1 2 のいずれか 1 項に記載のアプリケーションプログラム。

【0 1 2 2】

（付記 1 4） 前記画面テーマリソースファイルから前記画面テーマ定義データを読み出すのに代えて、前記画面テーマリソースファイルから読み出され、オペレーティングシステムのために確保されたメモリ領域に展開されている前記画面テーマ定義データを読み出す処理を前記コンピュータに実行させるように構成されることを特徴とする、付記 1 0 ～ 1 3 のいずれか 1 項に記載のアプリケー

ションプログラム。

【 0 1 2 3 】

（付記 1 5） オペレーティングシステムによって管理される画面テーマに関する情報が書き換えられたときに前記オペレーティングシステムから発せられる画面テーマ変更通知を受けた場合に、前記切替時画面テーマ読出処理ステップ及び前記 G U I 画面描画処理ステップを前記コンピュータに実行させるように構成されることを特徴とする、付記 1 0 ～ 1 4 のいずれか 1 項に記載のアプリケーションプログラム。

【 0 1 2 4 】

（付記 1 6） 同種のグラフィカル・ユーザ・インターフェース（以下、G U I という）部品を複数個含む G U I 画面の画面テーマを切り替える画面テーマ切替処理をコンピュータに実行させるためのアプリケーションプログラムを格納したコンピュータ読取可能な記憶媒体であって、

前記画面テーマ切替処理が、

切替後の画面テーマの画面テーマ定義データを画面テーマリソースファイルから読み出し、前記アプリケーションプログラムのために確保されたメモリ領域に記憶させる切替時画面テーマ読出処理ステップと、

前記アプリケーションプログラムのメモリ領域に記憶されている画面テーマ定義データを用いて前記 G U I 画面に含まれる同種の複数個の G U I 部品を描画する G U I 画面描画処理ステップとを含むことを特徴とする、アプリケーションプログラムを格納したコンピュータ読取可能な記憶媒体。

【 0 1 2 5 】

（付記 1 7） 前記アプリケーションプログラムが、前記切替時画面テーマ読出処理ステップにおいて、前記画面テーマリソースファイルから前記 G U I 画面に含まれる G U I 部品の画面テーマ定義データのみを読み出し、前記アプリケーションプログラムのメモリ領域に記憶させる処理を前記コンピュータに実行させるように構成されることを特徴とする、付記 1 6 記載のアプリケーションプログラムを格納したコンピュータ読取可能な記憶媒体。

【 0 1 2 6 】

（付記 1 8） 前記アプリケーションプログラムが、

起動時に、デフォルトとして設定された画面テーマの画面テーマ定義データを前記画面テーマリソースファイルから読み出し、前記アプリケーションプログラムのメモリ領域に記憶させるアプリ起動時画面テーマ読出処理ステップを含む処理を前記コンピュータに実行させるように構成され、

前記切替時画面テーマ読出処理ステップにおいて、切替後の画面テーマの画面テーマ定義データを前記画面テーマリソースファイルから読み出し、前記アプリケーションプログラムのメモリ領域に記憶されている画面テーマ定義データを更新する処理を前記コンピュータに実行させるように構成されることを特徴とする、付記 1 6 又は 1 7 記載のアプリケーションプログラムを格納したコンピュータ読取可能な記憶媒体。

【 0 1 2 7 】

（付記 1 9） 前記アプリケーションプログラムが、前記切替時画面テーマ読出処理ステップにおいて、オペレーティングシステムによって管理される画面テーマに関する情報を用いて前記画面テーマリソースファイルから画面テーマ定義データを読み出す処理を前記コンピュータに実行させるように構成されることを特徴とする、付記 1 6 ～ 1 8 のいずれか 1 項に記載のアプリケーションプログラムを格納したコンピュータ読取可能な記憶媒体。

【 0 1 2 8 】

（付記 2 0） 前記アプリケーションプログラムが、前記画面テーマリソースファイルから前記画面テーマ定義データを読み出すのに代えて、前記画面テーマリソースファイルから読み出され、オペレーティングシステムのために確保されたメモリ領域に展開されている前記画面テーマ定義データを読み出す処理を前記コンピュータに実行させるように構成されることを特徴とする、付記 1 6 ～ 1 9 のいずれか 1 項に記載のアプリケーションプログラムを格納したコンピュータ読取可能な記憶媒体。

【 0 1 2 9 】

（付記 2 1） 前記アプリケーションプログラムが、オペレーティングシステムによって管理される画面テーマに関する情報が書き換えられたときに前記オ

ペレーティングシステムから発せられる画面テーマ変更通知を受けた場合に、前記切替時画面テーマ読出処理ステップ及び前記 G U I 画面描画処理ステップを前記コンピュータに実行させるように構成されることを特徴とする、付記 1 6 ～ 2 0 のいずれか 1 項に記載のアプリケーションプログラムを格納したコンピュータ読取可能な記憶媒体。

【 0 1 3 0 】

【発明の効果】

以上詳述したように、本発明の携帯端末及びアプリケーションプログラムによれば、アプリケーションの画面が全体的に統一されたテーマ（画面テーマ）で表現され、G U I 画面の画面テーマを高速で切り替えることができるという利点がある。また、できるだけ少ないメモリ領域を使用しながら、G U I 画面の画面テーマの切替（特に高速切替）を実現できるという利点もある。

【図面の簡単な説明】

【図 1】

本発明の一実施形態にかかる携帯電話の全体構成を示す模式図である。

【図 2】

本発明の一実施形態にかかる携帯電話におけるアプリ起動時のアプリの G U I 画面の描画処理について説明するためのシーケンス図である。

【図 3】

本発明の一実施形態にかかる携帯電話における画面テーマ設定時の画面テーマリストの表示処理について説明するためのシーケンス図である。

【図 4】

本発明の一実施形態にかかる携帯電話における画面テーマ設定時の画面テーマ選択時、決定時の処理について説明するためのシーケンス図である。

【図 5】

本発明の一実施形態にかかる携帯電話における画面テーマ設定時の画面テーマ選択時、決定時の処理の他の例について説明するためのシーケンス図である。

【図 6】

本発明の一実施形態にかかる携帯電話におけるアプリの G U I 画面の画面テ

マ切替時の処理について説明するためのシーケンス図である。

【図 7】

本発明の一実施形態にかかる携帯電話において表示される画面テーマリストの例を示す図である。

【図 8】

本発明の一実施形態にかかる携帯電話において表示されるアプリの G U I 画面の例を示す図である。

【図 9】

本発明の一実施形態にかかる携帯電話において表示される画面テーマリストの選択位置を変えた場合の例を示す図である。

【図 1 0】

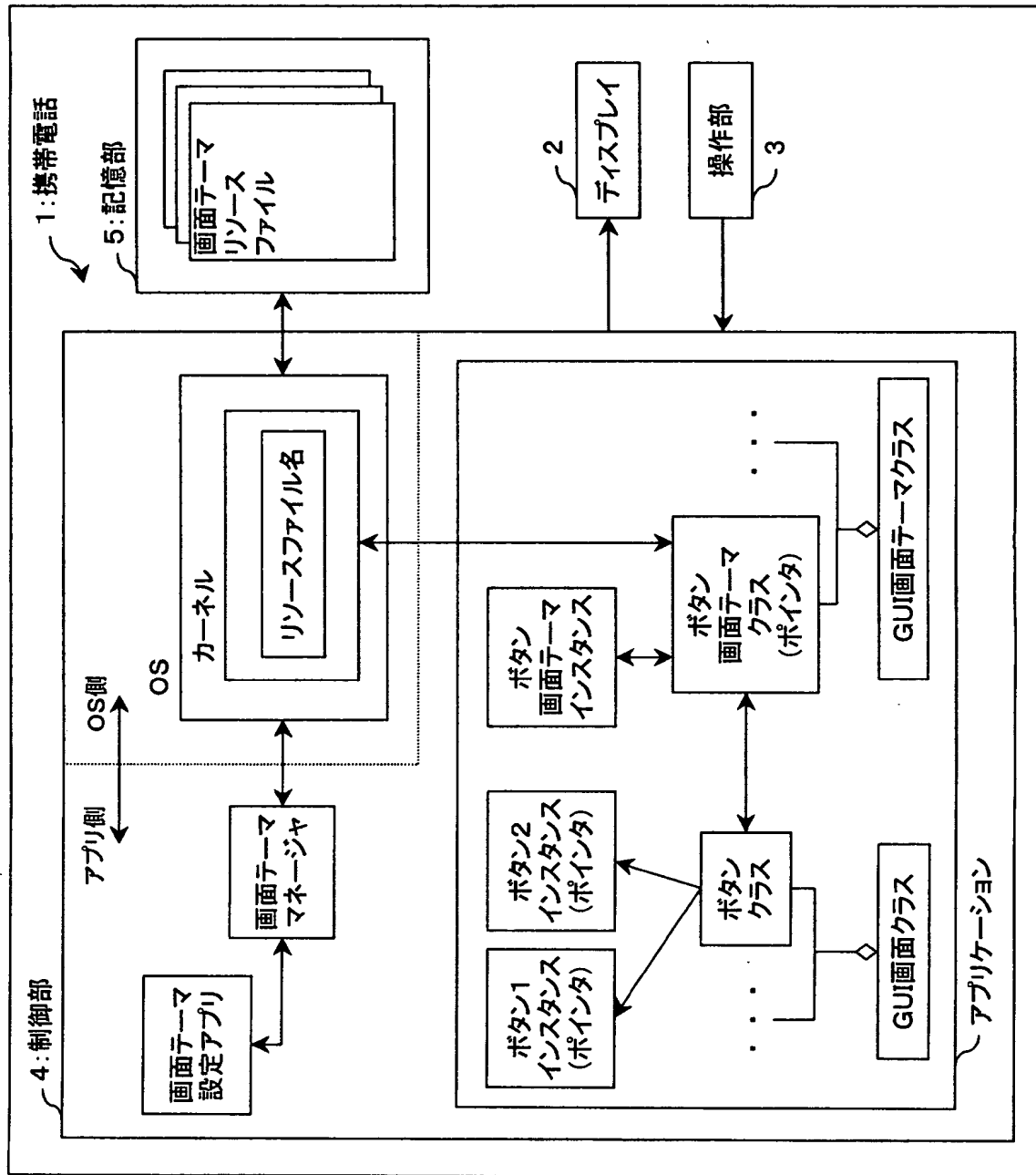
本発明の一実施形態にかかる携帯電話において表示されるアプリの G U I 画面の画面テーマが切り替えられた場合の例を示す図である。

【符号の説明】

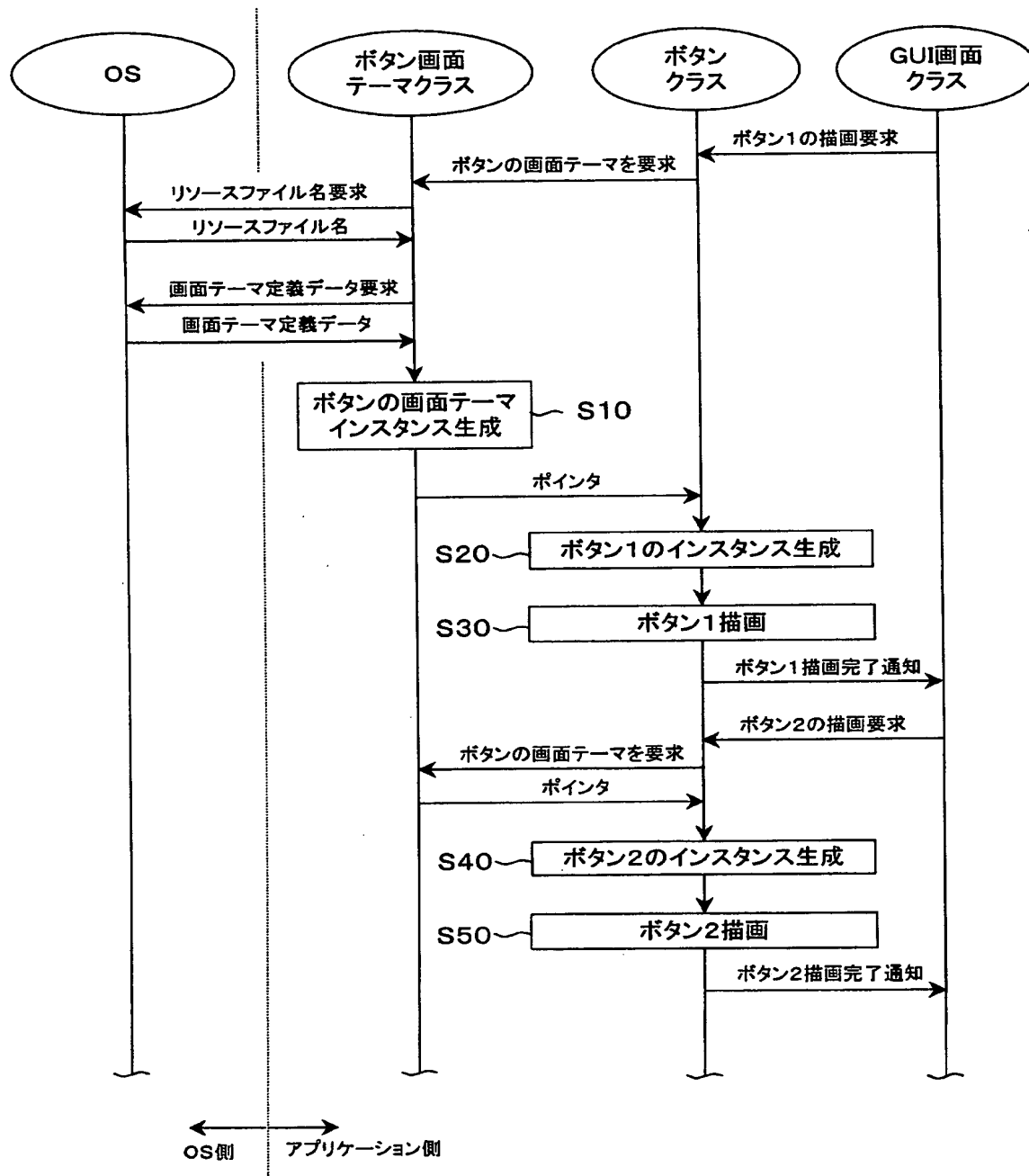
- 1 携帯電話（携帯端末）
- 2 ディスプレイ
- 3 操作部
- 4 制御部
- 5 記憶部

【書類名】 図面

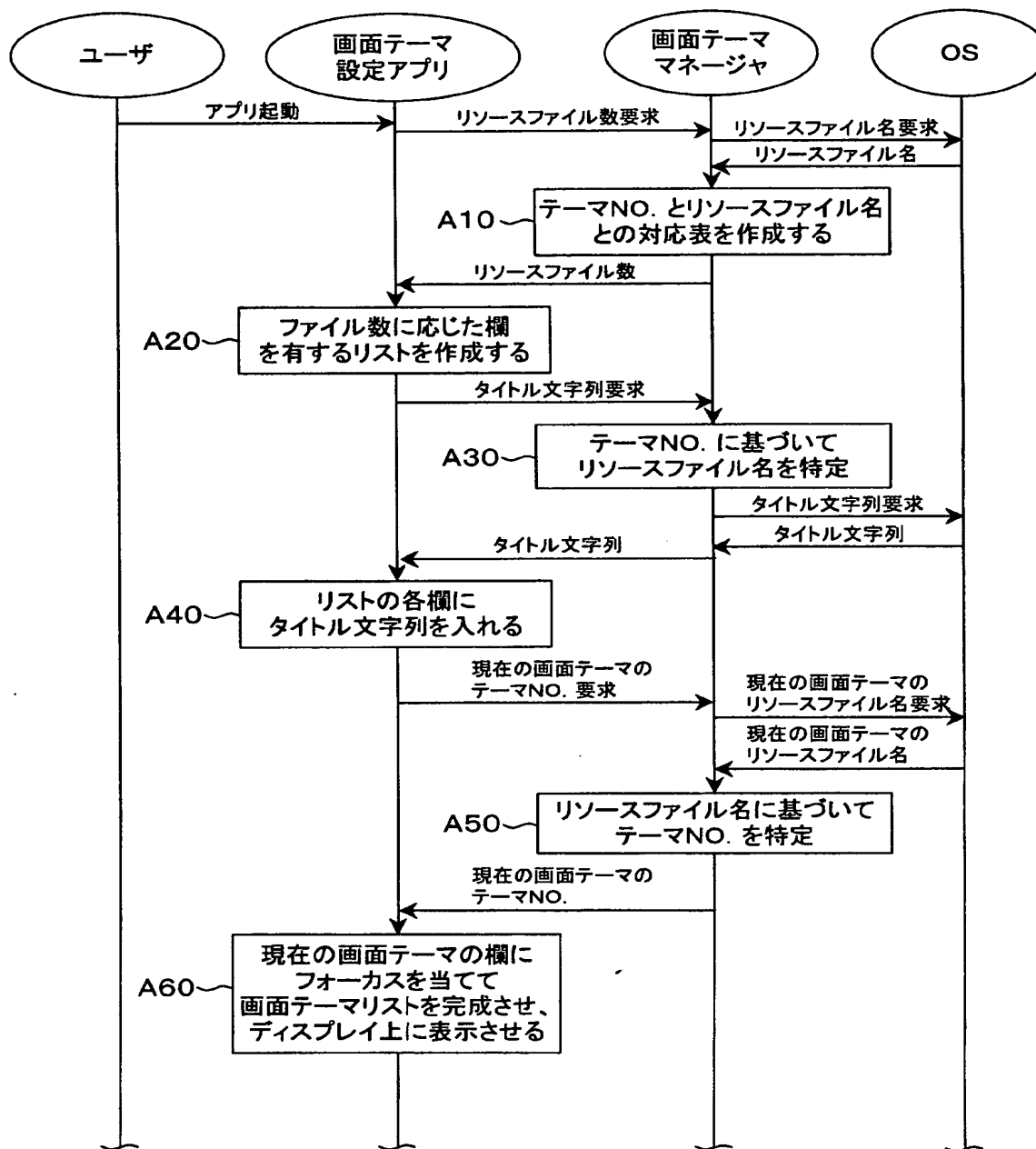
【図 1】



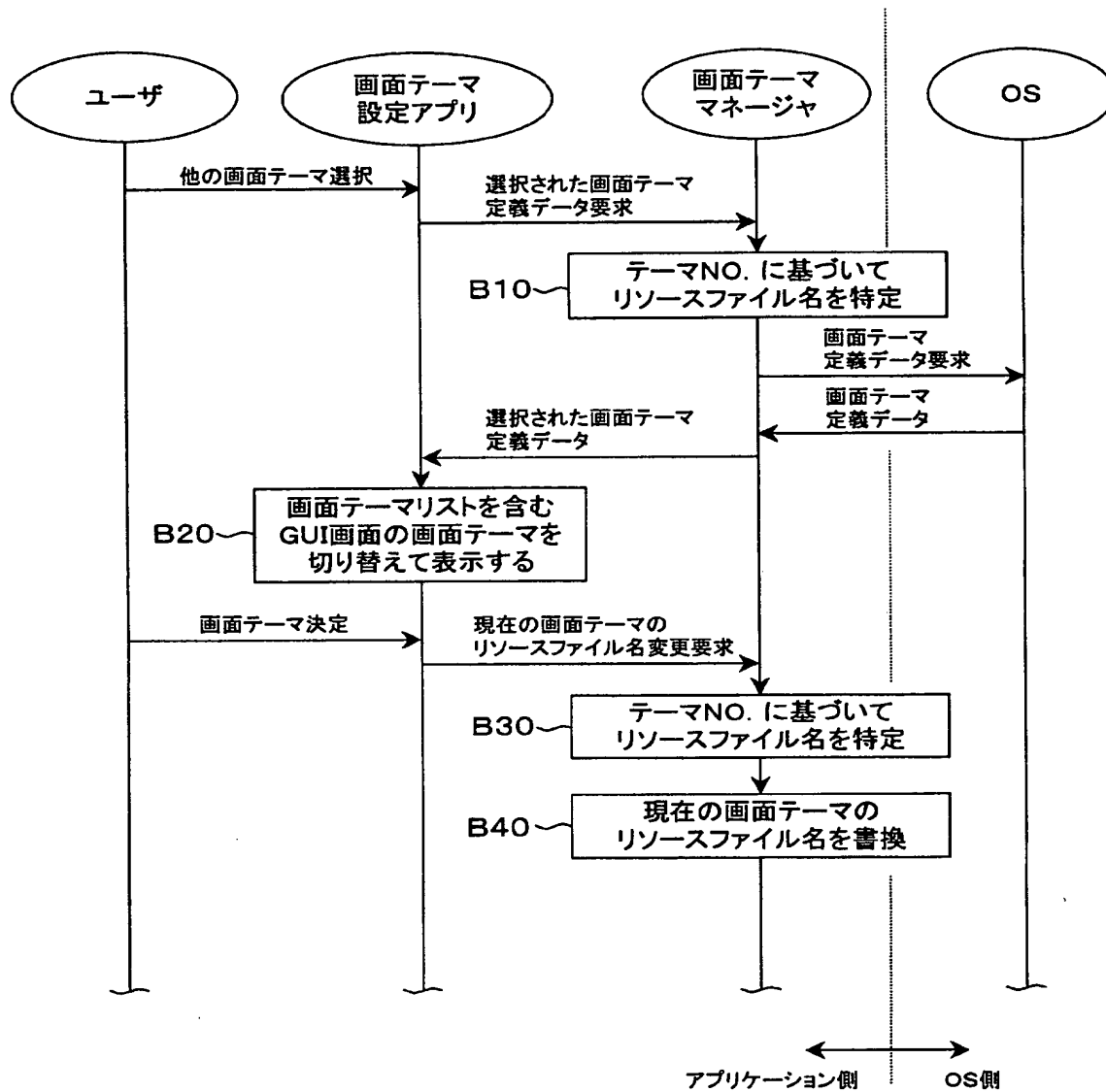
【図 2】



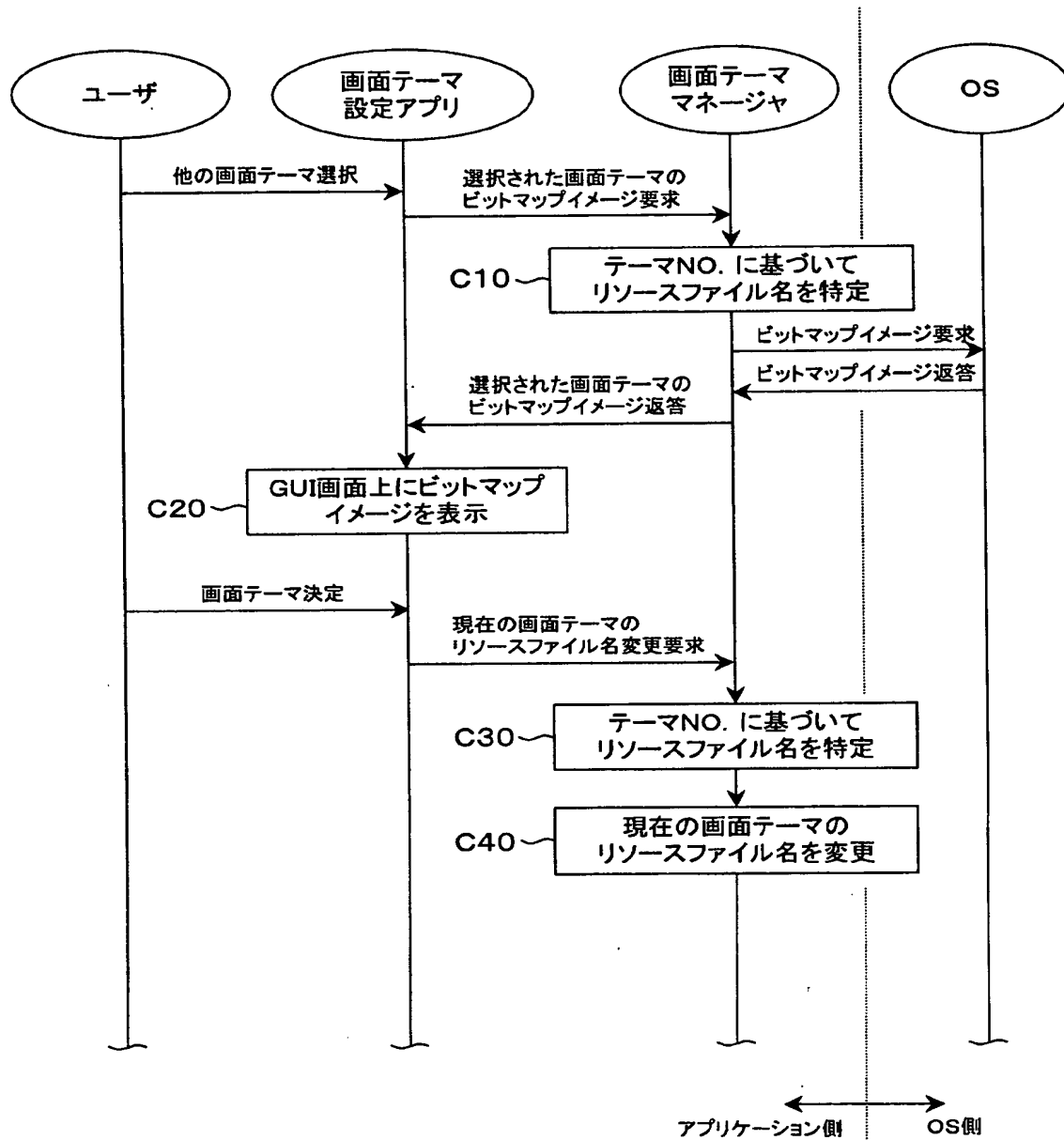
【図 3】



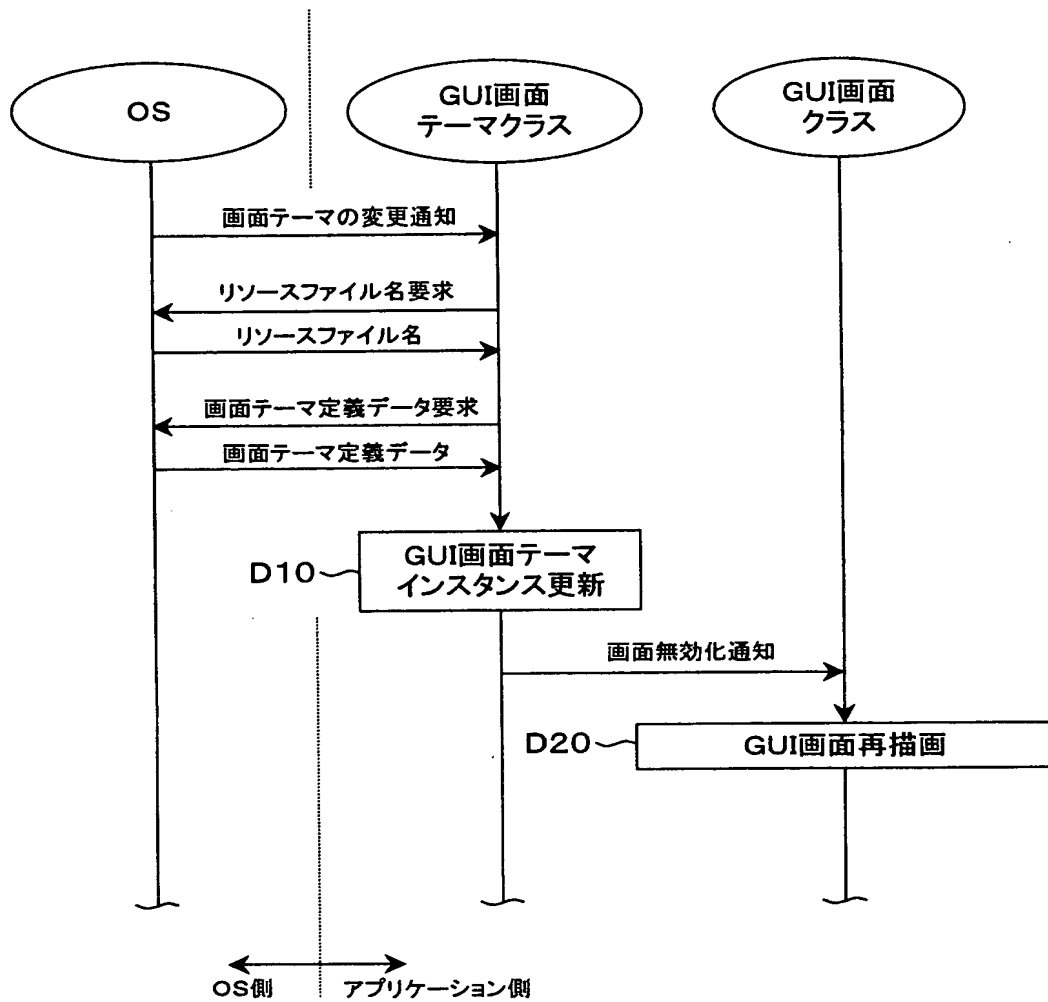
【図 4】



【図5】



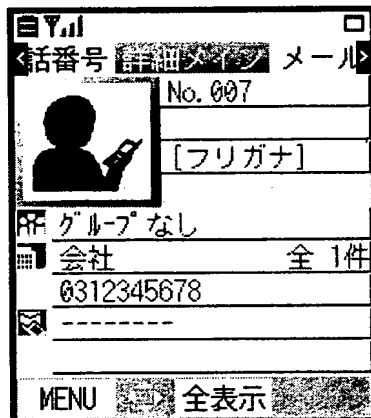
【図 6】



【図 7】



【図 8】



【図 9】



【図 1 0】



【書類名】 要約書

【要約】

【課題】 アプリの G U I 画面が全体的に統一された画面テーマで表現されるようにし、省メモリ化を図りながら、画面テーマが高速で切り替わるようにする。

【解決手段】 同種の G U I 部品を複数個含む G U I 画面を表示させるアプリの G U I 画面の画面テーマを切り替える処理を実行する制御部 4 と、画面テーマを定義する画面テーマ定義データを格納する記憶部とを備え、制御部 4 を、切替後の画面テーマの画面テーマ定義データを記憶部から読み出し、アプリのメモリ領域に記憶させる処理と、アプリのメモリ領域に記憶されている画面テーマ定義データを用いて G U I 画面に含まれる同種の複数個の G U I 部品を描画する処理とを実行するように構成する。

【選択図】 図 1



出 願 人 履 歴 情 報

識別番号 [000005223]

1. 変更年月日	1996年 3月26日
[変更理由]	住所変更
住 所	神奈川県川崎市中原区上小田中4丁目1番1号
氏 名	富士通株式会社